Patent
256/145
2021039-7011112001

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the Application of:	Group Art Unit: 2157
Sashikanth Chandrasekaran, et al.	Examiner: Avi M. Gold
Serial No.: 09/872,891	
Filed: May 31, 2001	
For: METHOD AND MECHANISM FOR PREDICTING DATA CONFLICTS AND GENERATING A LOAD DISTRIBUTION PLAN IN A MULTI-NODE SYSTEM	

APPEAL BRIEF UNDER 37 CFR § 41.37

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Applicant submits this Appeal Brief pursuant to the Notice of Appeal filed February 21, 2006. This brief is submitted in triplicate.

I. REAL PARTY IN INTEREST

The real party in interest is the assignee Oracle International Corporation of Redwood Shores, California.

II. RELATED APPEALS AND INTERFERENCES

To the best of Applicant's knowledge, there are no related appeals or interferences.

III. STATUS OF CLAIMS

Claims 1-66 are pending. Claims 1-66 are rejected, and are appealed.

IV. STATUS OF AMENDMENTS

No amendments have been filed after the final rejection dated November 21, 2005.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The present section of the Appeal Brief is set forth merely to comply with the requirements of 37 C.F.R. § 41.37(c)(v) and is not intended to limit the pending claims in any way.

Claim 1 recites "A method for predicting the behavior of a workload across a plurality of nodes, the method comprising: (a) receiving a workload to be executed; <u>(b) executing the workload on a single node before the workload is sent to a plurality of nodes for execution; (c) tracing the execution of the workload to identify a potential data conflict</u>; (d) based on a result of the tracing, predicting the behavior of the workload across the plurality of nodes; and (e) outputting the prediction." (Emphasis Added)

Claim 14 recites "A method for distributing a workload across a plurality of nodes, the method comprising: (a) receiving a workload to be executed; <u>(b) executing the workload on a single node before the workload is sent to a plurality of nodes for execution; (c) tracing the execution of the workload to identify a potential data conflict</u>; (d) based on a result of the tracing, forming a workload distribution scheme that distributes the workload across the plurality of nodes; and (e) outputting the workload distribution scheme." (Emphasis Added)

Claim 32 recites "A computer program product that includes a medium usable by a processor, the medium comprising a sequence of instructions which, when executed by said processor, causes said processor to execute a process for optimizing the distribution of a workload across a plurality of nodes, the process comprising: (a) receiving a workload to be executed; <u>(b) executing the workload on a single node before the workload is sent to a plurality of nodes for execution; (c) tracing the execution of the workload to identify a potential data conflict</u>; (d) based on a result of the tracing, optimizing the distribution of the workload across the plurality of nodes; and (e) outputting the optimized distribution scheme." (Emphasis Added)

Claim 33 recites "A computer program product that includes a medium usable by a processor, the medium comprising a sequence of instructions which, when executed by said processor, causes said processor to execute a process for distributing a workload across a plurality of nodes, the process comprising: (a) receiving a workload to be executed; <u>(b) executing the workload on a single node before the workload is sent to a plurality of nodes for execution; (c) tracing the execution of the workload to identify a potential data conflict</u>; (d) based on a result of the tracing, forming a workload distribution scheme that distributes the workload across the plurality of nodes; and (e) outputting the workload distribution scheme." (Emphasis Added)

Claim 34 recites "A system for distributing a workload across a plurality of nodes, comprising: (a) means for receiving a workload to be executed; <u>(b) means for executing the workload on a single node before the workload is sent to a plurality of nodes for execution; (c) means for tracing the execution of the workload to identify a potential data conflict</u>; (d) means for, based on a result of the tracing, forming a workload distribution scheme that distributes the workload

across the plurality of nodes; and (e) means for outputting the workload distribution scheme." (Emphasis Added)

Claim 35 recites "A system for optimizing the distribution of a workload across a plurality of nodes, comprising: (a) means for receiving a workload to be executed; <u>(b) means for executing the workload on a single node before the workload is sent to a plurality of nodes for execution; (c) means for tracing the execution of the workload to identify a potential data conflict</u>; (d) means for optimizing the distribution of the workload across the plurality of nodes based on a result of the tracing; and (e) means for outputting the optimized distribution scheme." (Emphasis Added)

Claim 36 recites "A computer program product that includes a medium usable by a processor, the medium comprising a sequence of instructions which, when executed by said processor, causes said processor to execute a process for predicting the behavior of a workload across a plurality of nodes, the process comprising: (a) receiving a workload to be executed; <u>(b) executing the workload on a single node before the workload is sent to a plurality of nodes for execution; (c) tracing the execution of the workload to identify a potential data conflict</u>; (d) based on a result of the tracing, predicting the behavior of the workload across the plurality of nodes; and (e) outputting the prediction." (Emphasis Added)

Claim 40 recites "A system for predicting the behavior of a workload across a plurality of nodes, comprising: (a) means for receiving a workload to be executed; <u>(b) means for executing the workload on a single node before the workload is sent to a plurality of nodes for execution; (c) means for tracing the execution of the workload to identify a potential data conflict</u>; (d) means for, based on a result of the tracing, predicting the behavior of the workload across the plurality of nodes; and (e) means for outputting the prediction." (Emphasis Added)

Claim 54 recites "A method for optimizing the distribution of a workload across a plurality of nodes, the method comprising: (a) receiving a workload to be executed; <u>(b) executing the workload on a single node before the workload is sent to a plurality of nodes for execution; (c) tracing the execution of the workload to identify a potential data conflict</u>; (d) based on a result of the

tracing, optimizing the distribution of the workload across the plurality of nodes; and (e) outputting the optimized distribution scheme." (Emphasis Added)

An example of a method that includes "executing the workload on a single node before the workload is sent to a plurality of nodes for execution," and "tracing the execution of the workload to identify a potential data conflict," is described in paragraphs 20-26 and figure 3 (see reference numbers 302, 304) of the specification. An example of a method that includes "outputting optimized distribution scheme" is described in paragraphs 31-35, and figure 4 of the specification.

A method that includes "executing the workload on a single node before the workload is sent to a plurality of nodes for execution" allows the execution of the workload in a multi-node environment be simulated on the single node. This has the advantageous of predicting a behavior of the execution of the workload in the multi-node environment before the workload is sent to the multi-node environment for actual execution. For example, if a result of the simulation indicates that it is desirable to execute the workload in a multi-node environment, then the workload may be sent to the multi-node environment for execution by the multi-nodes. On the other hand, if a result of the simulation indicates that it is not desirable to execute the workload in a multi-node environment, then the workload may be executed by the single node, and may not be sent to the multi-node environment for execution.

VI. ISSUES

The issues for this appeal are whether claims 1, 14, 32-36, 40, and 54 are patentable under 35 U.S.C. § 102(b) over U.S. Patent No. 5,657,450 ("Rao"), and whether claims 1-9, 11, 14-58, 60-63, 65, and 66 are patentable under 35 U.S.C. 103(a) over U.S. Patent No. 6,405,257 ("Gersht") in view of Rao.

VII. ARGUMENTS

A. Claim Rejections under 35 U.S.C. § 102

Claims 1, 14, 32-36, 40, and 54

Applicants submit that claims 1, 14, 32-36, 40, and 54 are patentable over Rao because Rao does not disclose or suggest at least two of the limitations recited in these claims.

Claims 1, 14, 32-36, 40, and 54 recite "executing the workload on a single node before the workload is sent to a plurality of nodes for execution." Rao does not disclose or suggest such limitation. Rather Rao discloses a method for providing a time estimate for processing a user request (column 1, lines 23-32, and column 1, line 65 to column 2, line 4). In Rao, the method of obtaining time estimate includes (1) receiving a request at an intermediary server, (2) analyzing the request to create sub-operations for obtaining information from different information sources, and (3) obtaining a time estimate for each information source (column 1, line 65 to column 2, line 52). As such, in Rao, the intermediary server merely provides time estimate for a request, and does not execute the workload on a single node before the workload is sent to a plurality of nodes for execution.

According to the Office Action, column 2, lines 40-45 allegedly disclose executing the workload on a single node before the workload is sent to a plurality of nodes for execution. However, the cited passage actually discloses:

> The method of the present invention is carried out on the intermediary server and is comprised of the following steps: receive an access operation directed to multiple distal information sources, analyze the requested operation to create sub-operations for each information source and to identify merge information for the operation, said merge information specifying when and in what form results are provided back to the client workspace; retrieve the model information for each information source which does not provide a completion time estimate. . .

As such, the cited passage of Rao discloses an intermediary server for determining sub-operations for each information source (i.e., to obtain the requested information from each information source),

and does not disclose or suggest executing a workload on a single node before the workload is sent to a plurality of nodes for execution, as recited in claims 1, 14, 32-36, 40, and 54.

For at least the foregoing reasons, Applicant respectfully submits that claims 1, 14, 32-36, 40, and 54, and their respective dependent claims, are patentable over Rao under 35 U.S.C. § 102.

Claims 1, 14, 32-36, 40, and 54 also recite "tracing the execution of the workload to identify a potential data conflict." Rao also does not disclose or suggest this second limitation. According to the Office Action, column 2, lines 1-53 allegedly disclose the above limitation. However, the cited passage actually discloses:

> A user operating in a client workspace gains access to multiple information sources through an intermediary server that is "close" to a client workspace (i.e. where the latencies and delays between the two are predictable and short). The client workspace and the intermediary server communicate using a generic protocol. The intermediary server in turn communicates with the various information servers using a protocol supported by the information source. The intermediary server synthesizes or otherwise obtains time and progress estimates responsive to such user requests. For operations involving multiple information sources, <u>time estimates are obtained for each information source and then synthesized to obtain time estimates for the operation</u>. Such estimates are then available to the client workspace for reporting back to the user.
>
> When compared to known systems which provide time estimates and progress feedback, the environment in which the present invention operates has a number of complications. In particular, the intermediary server has to deal with an open-ended set of disparate information sources at varying degrees of reliability and distance and usually outside of local control. Providing estimates is further complicated by the fact that an operation can span across multiple information sources. Merging the results received back from the multiple information sources is performed according to a query type and underlying merge timing policy. The time synthesized accounts for the query type and underlying merge timing policy.
>
> During the course of interaction with information sources that do not provide time estimates, the intermediary server builds up a timing model for each information source of how long various operations take to execute. The model will take into account a variety of factors including

> network distance and the hour of day. The models are used to create completion time or first response estimates for performing various operations with the information sources.
>
> The method of the present invention is carried out on the intermediary server and is comprised of the following steps: receive an access operation directed to multiple distal information sources, analyze the requested operation to create sub-operations for each information source and to identify merge information for the operation, said merge information specifying when and in what form results are provided back to the client workspace; retrieve the model information for each information source which does not provide a completion time estimate; <u>generate completion time estimates for each sub-operation</u> to each such information source based on their model; for information sources that do provide estimates, <u>issue a set-up operation to obtain an estimate</u>, factor in merge processing costs with selected time estimate <u>to create time estimate</u>; provide time estimate to the client workspace; and updating the model for information sources with new actual timing information.

As such, Rao discloses obtaining time estimates to perform an operation. There is nothing in the cited passage of Rao that discloses or suggests potential data conflict, much less, identifying a potential data conflict by tracing an execution of workload.

Also according to the Office Action, Rao discloses obtaining time estimates, which allegedly involves determining data conflicts. Applicant respectfully disagree. Determining a potential data conflict is not an inherent step in obtaining a time estimate for performing an operation. As such, a mere disclosure of determining time estimates does not necessitate a finding that a determination of potential data conflict is disclosed. In addition, Rao teaches two techniques for obtaining time estimates, neither of which involves determining a potential data conflict. In particular, Rao discloses obtaining a time estimate directly from an information source that provides a time estimate for a particular operation (column 2, lines 47-50, and column 8, lines 23-30). On the other hand, if the information source does not provide a time estimate, a timing model is used to obtain a time estimate (column 2, lines 43-45, and column 8, lines 30-41). As such, Rao does not disclose or suggest determining a potential data conflict, much less, determining a potential data conflict by tracing an execution of workload.

For these additional reasons, Applicant respectfully submits that claims 1, 14, 32-36, 40, and 54, and their respective dependent claims, are patentable over Rao under 35 U.S.C. § 102.

B. Claim Rejections under 35 U.S.C. § 103

Claims 1, 14, 32-36, 40, and 54

Applicant submits that claims 1, 14, 32-36, 40, and 54 are patentable over Gersht and Rao because the combination of these references does not disclose or suggest at least two of the limitations recited in these claims.

Claims 1, 14, 32-36, 40, and 54 recite "executing the workload on a single node before the workload is sent to a plurality of nodes for execution." Gersht does not disclose or suggest such limitation. Rather Gersht discloses a source node in a network that controls the external traffic directed to the source node based on a set of requirements (column 1, line 60 to column 2, line 41). According to page 5 the Office Action, column 2, lines 10-16 of Gersht allegedly disclose the above limitation[1]. However, the cited passage of Gersht actually discloses:

> Each source node in the network then controls, at the burst level, the external traffic directed to the source node based on the preallocated set of maximum permitted rates and the preassigned set of burst access thresholds and without communicating with other nodes in the network. Specifically, each source node detects the beginning of a burst in the traffic that is accessing the source node.

As such, the cited passage of Gersht discloses a source node that controls network traffic, and does not disclose or suggest executing the workload on a single node before the workload is sent to a plurality of nodes for execution.

[1] Applicant notes that on page 6 of the Office Action, the Examiner appears to agree that Gersht does not disclose "executing the workload on a single node before the workload is sent to a plurality of nodes for execution." This appears to be contradictory with the Examiner's statement on page 5 of the Office Action, where column 2, lines 10-16 of Gersht are cited for the alleged disclosure of the limitation. For the sake of completeness for this brief, Applicant hereby addresses the Examiner's statement on page 5 of the Office Action.

Claims 1, 14, 32-36, 40, and 54 also recite "tracing the execution of the workload to identify a potential data conflict." Gersht also does not disclose or suggest this second limitation." According to the Office Action, column 2, lines 15-20 of Gersht allegedly disclose the above limitation. However, the cited passage of Gersht actually discloses:

> Specifically, each source node detects the beginning of a burst in the traffic that is accessing that source node. From the predetermined set of routes between the source-destination node pair and <u>based on the peak packet rate, service class, and quality of service (Qos) requirements of the detected burst</u>, the source node identifies a predetermined route for the burst.

As such, the cited passage of Gersht specifically teaches determining a route for a burst based on "peak packet rate, service class, and quality of service (Qos) requirements," none of which is a potential data conflict. In fact, there is nothing in Gersht that discloses or suggests identifying a potential data conflict, much less, identifying a potential data conflict by tracing the execution of a workload.

As discussed, Rao also does not disclose or suggest "executing the workload on a single node before the workload is sent to a plurality of nodes for execution," and "tracing the execution of the workload to identify a potential data conflict," and therefore, fails to make up the deficiencies present in Gersht. Since both Rao and Gersht fail to disclose the above limitations, they cannot be combined to form the resulting subject matter.

For at least the foregoing reasons, Applicant respectfully submits that claims 1, 14, 32-36, 40, and 54, and their respective dependent claims, are patentable over Gersht and Rao under 35 U.S.C. § 103.

VIII. CONCLUSION

For the above reasons, Applicant respectfully submits that rejection of claims 1-66 has been overcome. Accordingly, Applicant requests that the Board of Patent Appeals and Interferences overrule the Examiner and allow claims 1-66.

Respectfully submitted,

Bingham McCutchen LLP

Dated: May 19, 2006

By: Gerald Chan

Gerald Chan
Reg. No. 51,541

Three Embarcadero Center, Suite 1800
San Francisco, CA 94111-4067
Telephone: (650) 849-4904
Telefax: (650) 849-4800

APPENDIX A: Pending Claims

Listing of Appealed Claims 1-66.

1. (Previously Presented) A method for predicting the behavior of a workload across a plurality of nodes, the method comprising:

a) receiving a workload to be executed;

b) executing the workload on a single node before the workload is sent to a plurality of nodes for execution;

c) tracing the execution of the workload to identify a potential data conflict;

d) based on a result of the tracing, predicting the behavior of the workload across the plurality of nodes; and

e) outputting the prediction.

2. (Original) The method of claim 1 wherein the action of identifying potential data conflicts comprises predicting how many data conflicts will occur.

3. (Original) The method of claim 1 wherein the action of identifying potential data conflicts comprise predicting types of data conflicts.

4. (Original) The method of claim 3 in which the types of data conflicts comprises a read-write conflict.

5. (Original) The method of claim 3 in which the types of data conflicts are based upon types of operations needed to resolve the data conflicts.

6. (Original) The method of claim 3 in which the different types of data conflicts have differing levels of expense associated with operations needed for data conflict resolution.

7. (Original) The method of claim 1 in which the potential data conflicts are at the granularity of a data block.

8. (Original) The method of claim 1 in which the potential data conflicts are identified based upon workload division between sessions.

9. (Original) The method of claim 1 further comprising:

f) selecting a number of nodes;

g) dividing the traced execution of the workload across the number of nodes.

10. (Original) The method of claim 9 in which modulo division is used to divide the traced execution of the workload across the number of nodes.

11. (Original) The method of claim 9 in which the number of nodes corresponds to an anticipated number of nodes for a distributed computing system.

12. (Original) The method of claim 9 in which a modulo class represents a node in the number of nodes.

13. (Original) The method of claim 1 in which the potential data conflicts are used to compute costs of migrating the workload to a distributed system.

14. (Previously Presented) A method for distributing a workload across a plurality of nodes, the method comprising:

a) receiving a workload to be executed;

b) executing the workload on a single node before the workload is sent to a plurality of nodes for execution;

c) tracing the execution of the workload to identify a potential data conflict;

d) based on a result of the tracing, forming a workload distribution scheme that distributes the workload across the plurality of nodes; and

e) outputting the workload distribution scheme.

15. (Previously Presented) The method of claim 14, wherein the forming the workload distribution scheme comprises determining workload distribution in manner which reduces the potential data conflicts.

16. (Original) The method of claim 14, wherein the workload distribution scheme is based upon data accesses.

17. (Original) The method of claim 16 in which the workload is grouped in the workload distribution scheme to maximize intersection of data access on a same group of nodes.

18. (Original) The method of claim 16 in which the workload is grouped in the workload distribution scheme to minimize intersection of data access across different groups of nodes.

19. (Original) The method of claim 14, wherein the workload distribution scheme is based upon access frequencies.

20. (Original) The method of claim 19 in which data objects accessed by the workload are associated with weighting factors.

21. (Original) The method of claim 20 in which not all the data objects are associated with same weighting factors.

22. (Original) The method of claim 20 in which a weighted correlation is performed between the data objects and entities that access the data objects.

23. (Original) The method of claim 22 in which the entities that access the data objects comprises sessions.

24. (Original) The method of claim 22 in which subsets of the entities that access the data objects are grouped together.

25. (Original) The method of claim 24 in which a data structure is employed to represent an affinity between one of the entities that access the data objects and another of the entities.

26. (Original) The method of claim 14 in which the workload comprises data access upon one or more hierarchical objects.

27. (Original) The method of claim 26 in which tracing the execution of the workload comprises tracing identifiers for the one or more hierarchical objects.

28. (Original) The method of claim 14 in which tracing the execution of the workload comprises tracing identifiers associated with entities that access data.

29. (Original) The method of claim 28 in which the entities comprise sessions.

30. (Original) The method of claim 28 in which the workload distribution scheme distributes the workload based upon partitioning of the entities that access data.

31. (Previously Presented) The method of claim 30 in which an association is formed between partitioning of the entities that access data and partitioning of one or more applications within the workload.

32. (Previously Presented) A computer program product that includes a medium usable by a processor, the medium comprising a sequence of instructions which, when executed by said processor, causes said processor to execute a process for optimizing the distribution of a workload across a plurality of nodes, the process comprising:

a) receiving a workload to be executed;

b) executing the workload on a single node before the workload is sent to a plurality of nodes for execution;

c) tracing the execution of the workload to identify a potential data conflict;

d) based on a result of the tracing, optimizing the distribution of the workload across the plurality of nodes; and

e) outputting the optimized distribution scheme.

33. (Previously Presented) A computer program product that includes a medium usable by a processor, the medium comprising a sequence of instructions which, when executed by said processor, causes said processor to execute a process for distributing a workload across a plurality of nodes, the process comprising:

a) receiving a workload to be executed;

b) executing the workload on a single node before the workload is sent to a plurality of nodes for execution;

c) tracing the execution of the workload to identify a potential data conflict;

d) based on a result of the tracing, forming a workload distribution scheme that distributes the workload across the plurality of nodes; and

e) outputting the workload distribution scheme.

34. (Previously Presented) A system for distributing a workload across a plurality of nodes, comprising:

a) means for receiving a workload to be executed;

b) means for executing the workload on a single node before the workload is sent to a plurality of nodes for execution;

c) means for tracing the execution of the workload to identify a potential data conflict;

d) means for, based on a result of the tracing, forming a workload distribution scheme that distributes the workload across the plurality of nodes; and

e) means for outputting the workload distribution scheme.

35. (Previously Presented) A system for optimizing the distribution of a workload across a plurality of nodes, comprising:

a) means for receiving a workload to be executed;

b) means for executing the workload on a single node before the workload is sent to a plurality of nodes for execution;

c) means for tracing the execution of the workload to identify a potential data conflict;

d) means for optimizing the distribution of the workload across the plurality of nodes based on a result of the tracing; and

e) means for outputting the optimized distribution scheme.

36. (Previously Presented) A computer program product that includes a medium usable by a processor, the medium comprising a sequence of instructions which, when executed by said processor, causes said processor to execute a process for predicting the behavior of a workload across a plurality of nodes, the process comprising:

a) receiving a workload to be executed;

b) executing the workload on a single node before the workload is sent to a plurality of nodes for execution;

c) tracing the execution of the workload to identify a potential data conflict;

d) based on a result of the tracing, predicting the behavior of the workload across the plurality of nodes; and

e) outputting the prediction.

37. (Previously Presented) The computer program product of claim 36 wherein the action of identifying potential data conflicts comprises predicting how many data conflicts will occur.

38. (Previously Presented) The computer program product of claim 36 wherein the action of identifying potential data conflicts comprises predicting types of data conflicts.

39. (Previously Presented) The computer program product of claim 36 in which the potential data conflicts are identified based upon workload division between sessions.

40. (Previously Presented) A system for predicting the behavior of a workload across a plurality of nodes, comprising:

a) means for receiving a workload to be executed;

b) means for executing the workload on a single node before the workload is sent to a plurality of nodes for execution;

c) means for tracing the execution of the workload to identify a potential data conflict;

d) means for, based on a result of the tracing, predicting the behavior of the workload across the plurality of nodes; and

e) means for outputting the prediction.

41. (Previously Presented) The system of claim 40 wherein the means for tracing is configured to predict how many data conflicts will occur.

42. (Previously Presented) The system of claim 40 wherein the means for tracing is configured to predict types of data conflicts.

43. (Previously Presented) The system of claim 40 in which the means for tracing is configured to identify the potential data conflicts based upon workload division between sessions.

44. (Previously Presented) The computer program product of claim 33, wherein the forming the workload distribution scheme comprises determining workload distribution in manner which reduces the potential data conflicts.

45. (Previously Presented) The computer program product of claim 33, wherein the workload distribution scheme is based upon data accesses.

46. (Previously Presented) The computer program product of claim 33, wherein the workload distribution scheme is based upon access frequencies.

47. (Previously Presented) The computer program product of claim 33 in which the workload comprises data access upon one or more hierarchical objects.

48. (Previously Presented) The computer program product of claim 33 in which tracing the execution of the workload comprises tracing identifiers associated with entities that access data.

49. (Previously Presented) The system of claim 34, wherein the means for forming the workload distribution scheme comprises means for determining workload distribution in manner which reduces the potential data conflicts.

50. (Previously Presented) The system of claim 34, wherein the workload distribution scheme is based upon data accesses.

51. (Previously Presented) The system of claim 34, wherein the workload distribution scheme is based upon access frequencies.

52. (Previously Presented) The system of claim 34 in which the workload comprises data access upon one or more hierarchical objects.

53. (Previously Presented) The system of claim 34 in which the means for tracing the execution of the workload comprises means for tracing identifiers associated with entities that access data.

54. (Previously Presented) A method for optimizing the distribution of a workload across a plurality of nodes, the method comprising:

a) receiving a workload to be executed;

b) executing the workload on a single node before the workload is sent to a plurality of nodes for execution;

c) tracing the execution of the workload to identify a potential data conflict;

d) based on a result of the tracing, optimizing the distribution of the workload across the plurality of nodes; and

e) outputting the optimized distribution scheme.

55. (Previously Presented) The method of claim 54, wherein the action of identifying potential data conflicts comprises predicting how many data conflicts will occur.

56. (Previously Presented) The method of claim 54, wherein the action of identifying potential data conflicts comprise predicting types of data conflicts.

57. (Previously Presented) The method of claim 54 in which the potential data conflicts are at the granularity of a data block.

58. (Previously Presented) The method of claim 54 in which the potential data conflicts are identified based upon workload division between sessions.

59. (Previously Presented) The method of claim 54 in which the potential data conflicts are used to compute costs of migrating the workload to a distributed system.

60. (Previously Presented) The computer program product of claim 32, wherein the action of identifying potential data conflicts comprises predicting how many data conflicts will occur.

61. (Previously Presented) The computer program product of claim 32, wherein the action of identifying potential data conflicts comprise predicting types of data conflicts.

62. (Previously Presented) The computer program product of claim 32 in which the potential data conflicts are at the granularity of a data block.

63. (Previously Presented) The computer program product of claim 32 in which the potential data conflicts are identified based upon workload division between sessions.

64. (Previously Presented) The computer program product of claim 32 in which the potential data conflicts are used to compute costs of migrating the workload to a distributed system.

65. (Previously Presented) The system of claim 35, wherein the means for tracing is configured to predict how many data conflicts will occur.

66. (Previously Presented) The system of claim 35, wherein the means for tracing is configured to predict types of data conflicts.